CYBAGE
Delivering Value. Scientifically.

Software Requirements Specification (SRS)
Investment Management Dashboard

Introduction

Purpose

The Investment Management Dashboard to be designed to help individual users track their personal investments across multiple categories such as Mutual Funds, Equity, and Debt.

The application will:

- Display a consolidated view of investments.
- Allow adding new investments.
- Show real-time portfolio summary based on API data.

Scope

- Frontend: Angular 20 SPA using latest features:
    1. **Signals API** for reactive state management.
    2. **Standalone Components** for modular architecture.
    3. **New Control Flow Syntax** (@if, @for, @switch).
- **UI**: Responsive design using Angular Material.
- **Backend**: REST API (mocked for training).
- Features:
    1. Dashboard with summary cards.
    2. Portfolio listing with Material Table.
    3. Add Investment form with validation.
    4. Lazy-loaded routes for Dashboard, Portfolio, Reports.
- Out of Scope:
    1. Authentication/Authorization.
    2. Advanced backend integration.

Objectives

- Provide a hands-on case study for Angular 20 features.
- Demonstrate API consumption and state management.
- Implement best practices for performance and maintainability.

Functional Requirements

ID	Requirement	Description	Acceptance Criteria
FR1	Portfolio Summary	Display Total Invested Amount and Current Value	Summary updates dynamically when investments change
FR2	Investment List	Show all investments in a Material Table	Table supports sorting and filtering
FR3	Add Investment	Form to add new investment	Form validates required fields and updates table
FR4	Edit/Delete Investment	Modify or remove investments	Changes reflect immediately in UI
FR5	Routing	Dashboard, Portfolio, Reports routes	Routes are lazy-loaded
FR6	State Management	Use Signals for reactive updates	No manual change detection required

Non-Functional Requirements

- **Performance:**
    1. Dashboard loads within 2 seconds on standard broadband.
- **Usability:**
    1. Responsive design for desktop and tablet.
    2. Consistent Material Design theme.
- **Maintainability:**
    1. Modular architecture using Standalone Components.
    2. Code follows Angular 20 best practices.
- **Reliability**:
    1. API errors handled gracefully with interceptors.
    2. Retry mechanism for failed requests.

System Architecture

- **Components**:
    1. DashboardComponent → Displays summary.
    2. PortfolioComponent → Lists investments.
    3. AddInvestmentComponent → Handles form input.
- **Data Flow**:
    1. HttpClient → API → Signals → Components.
- **State Management**:
    1. Signals for local state.
    2. NgRx for global state (optional advanced feature).

UI Mockups (Descriptive)

- Dashboard:
  - Two Material cards:
    - Total Invested: ₹XX,XXX
    - Current Value: ₹YY,YYY
- Portfolio Page:
  - Material Table:
    - Columns: Name | Type | Amount | Purchase Date | Current Value
    - Actions: Edit | Delete
- Add Investment Form:
  - Fields:
  - Name (Text)
  - Type (Dropdown: Equity, Debt, Mutual Fund)
  - Amount (Number)
  - Purchase Date (Date Picker)
  - Current Value (Number)
- Buttons: Save, Cancel

API Integration Details

- **Base URL**: /api/investments
- **Endpoints**:
  - GET /investments → Fetch all investments
    - Response:

```
{
  "investments": [
   { "id": 1, "name": "HDFC Mutual Fund", "type": "Mutual Fund", "amount": 5000,
"purchaseDate": "2025-11-17", "currentValue": 500 }
  ]
}
```

- POST /investments → Add new investment
- Request

```
{ "name": "ICICI", "type": "Mutual Fund", "amount": 5000, "purchaseDate": "2025-11-17",
"currentValue": 500 }
```

Angular Features Used:

- HttpClient for API calls.
- Interceptors for error handling.
- Signals for reactive state.
- Standalone Components for modular design.

Technology Stack

- Angular 20
- TypeScript
- Angular Material
- NgRx (optional advanced)
- ESBuild for optimization

Error Handling & Validation

- Form Validation:
  - Required fields: Name, Type, Amount, Purchase Date.
  - Amount > 0.
- API Errors:
  - Show Material Snackbar for failures.
  - Retry on network errors.